Assignment 1 Jack Sturtevant

Thursday, September 5, 2019 12:42 PM

1) $1XXX \rightarrow 8 \rightarrow 15$

$$Avg = \frac{\sum_{i=8}^{15}}{8} = \frac{8+9+10+11+12+13+14+15}{8} = \frac{92}{8} = 11.5$$

2) For a population of size n having l-bit strings, it can have exactly $n*2^l$ unique schemas only if the size of the population is 2 (n=2), and if the two string are 1's compliments of each other. This is because each string has exactly 2^l schemas associated with it. To add another 2^l schemas, the next member of the population must have exactly the opposite value in each bit position. If another member is added to the population, then each bit position will match one of the previous bits in the same position, making it impossible for this third member to have 2^l unique schemas. It follows that adding m more members will face the same problem

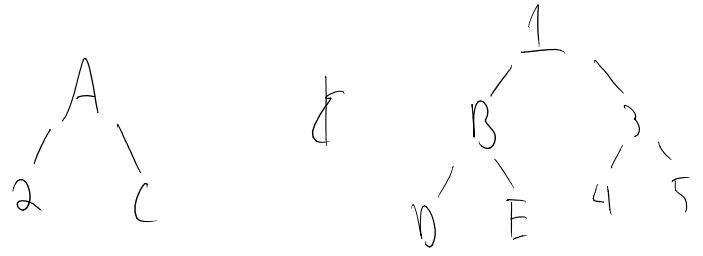
3) $$E[m(H,t+1)] \geq \frac{\bar{f}(H)}{\bar{f}} \left(1 - p_c \frac{d(H)}{L-1}\right)(1-p_m)^{O(H)} m(H,t)$$

$$\geq \frac{14}{20}\left(1-(0.5)\frac{5}{9}\right)(1-0.001)^2 * 3$$

$$\geq \frac{7}{10}\left(\frac{6.5}{9}\right)(0.998) * 3$$

$$\geq \boxed{1.5}$$

4)



5) Assuming that one point crossover, then there are 6 possible crossovers. There are two possible crossovers at the root, and two possible crossovers at each node that is a child of the root. This results in 2 + 2*(2) = 6.

6) $$child = \frac{[0+4, 5+-3, 6+-2, 1+-7]}{2} = [2.0, 1.0, 2.0, -3.0]$$

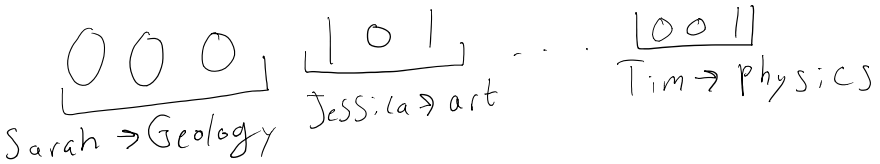
7) Mutations

8)

9) The chromosomes I made have a length of 18. Each librarian is encoded by 3 bits, which are a binary representation of numbers 0-5. 0 corresponds to geology, 1 to physics, etc… Sarah is denoted by the first 3 bits, Jessica by the second 3 bits, etc… This is depicted below:

000 101 ... 1001

3 bits, Jessica by the second 3 bits, etc… This is depicted below:



b)
The fitness function I used is simply the sum of the values associated with each librarian-category pair squared. By only taking the sum, there are many best values closely associated between 39-44 books per hour, so the algorithm doesn't add much more weight to the best solution, hence the squared. If there are any duplicate categories, or the value for a given 3-bit section is 6 or 7, then the fitness value is set to 0, because these are not possible. No helper functions were used, but a dictionary that is a property of the SGA class was made. This dictionary has keys which are the person's name, and values which are an internal dictionary. This internal dictionary associates the keys 0-7 to the corresponding value of a librarian sorting a category. For example, fitDict['Sarah'][3] would yield 6, because Sarah's history value is 6.

c)
The best solution produced by this algorithm yields 44 books sorted per hour. It has Sarah sorting Geology, Jessica sorting History, George sorting art, Karen sorting Chemistry, Sam sorting Poetry, and Tim sorting Physics. This solution is very rarely found by the algorithm, and most of the time it yields a solution that allows 39 books to be sorted per hour.

D)
Analytically, I determined that 44 books per hour is the optimal solution. However, for the genetic algorithm to achieve this solution, I had it run 2000 times, and record the best solution I found. I don't know how many times this solution was found, but I know that when I ran it 200 times, it did not produce this solution.

e)
It is absolutely possible for this algorithm to lose the optimal solution between generations. Because the next parents are determined probabilistically, and there can be mutations and crossovers, there is no guarantee the program will maintain the best solution. Especially when there are solutions with very close fitness values, with very different chromosome representations, it's possible that only the 2nd best (for example) fitness is passed on to the future generation. Or, there could be a cross over or mutation that eliminates the best solution, and is never passed forward because this change has a very low fitness.